359. A system comprising:

a first unit adapted to execute register-based instructions;

a hardware unit associated with the first unit, the hardware unit adapted to translate stack-based instructions into register-based instructions to be provided to the first unit for execution; and

a cache operably connected to the hardware unit, the cache adapted to provide at least some of the stack-based instructions to the hardware unit.

360. The system of claim 359, further comprising a system memory adapted to provide register-based instructions to the cache.

361. The system of claim 359, wherein the cache is an instruction cache.

362. The system of claim 359, wherein the first unit and the hardware unit are part of a central processing unit.

363. The system of claim 359, wherein the first unit comprises a central processing unit and wherein the hardware unit is outside of the central processing unit.

364. The system of Claim 359, wherein the first unit, hardware unit and cache are on the same chip.

365. The system of Claim 359, wherein the stack-based instructions are Java bytecodes.

366. The system of claim 359, wherein a portion of the operand stack is stored in a register file of the first unit.

367. The system of Claim 366, wherein the hardware unit produces register-based instructions that access the portion of the operand stack in the register file so as to reduce the number of register-based instructions that would otherwise be required.

368. The system of claim 366, wherein the hardware unit is adapted to produce at least one of overflow or underflow indications for the portion of the operand stack stored in the register file.

369. The system of claim 366, wherein an overflow or underflow produces operand transfer between the register file in the first unit and a memory.

370. The system of claim 368, wherein the registers of the register file of the central processing unit used to store the portion of operand stack is full of valid data.

D1 cont.

371. The system of claim 368, wherein the at least one of the overflow or underflow indications is generated by a stack instruction pushing an operand or popping the operand from the operand stack.

372. The system of claim 366, wherein the portion of the operand stack stored in the register file is maintained at or above a minimum level and/or at or below a maximum level.

373. The system of claim 366, wherein the hardware unit is further adapted to store at least some Java registers in the register file.

374. The system of claim 359, wherein the hardware unit implements at least part of a Java virtual machine.

375. The system of claim 359, wherein the hardware unit is connected between a the cache memory and the first unit.

376. The system of claim 359, wherein the hardware unit is adapted to manage a at least a portion of a Java stack.

377. The system of claim 359, wherein the hardware unit is adapted to examine the stack-based instructions to determine whether multiple stack-based instructions can be combined into fewer register-based instructions.

378. The system of claim 377, wherein the hardware unit produces register-based instructions that access a portion of the operand stack in the register file so as to reduce the number of register-based instructions that would otherwise be required.

379. The system of claim 377, wherein multiple stack-based instructions pass through the hardware unit concurrently to allow for the operation of combining logic.

380. The system of claim 359, wherein the hardware unit is adapted to convert multiple Java bytecodes into a single register-based instruction.

381. The system of claim 380, wherein the multiple Java bytecodes include a basic operand instruction and one or more stack manipulation instructions.

382. The system of claim 380, wherein the multiple Java bytecodes include a load or store instruction.

383. The system of claim 380, wherein the hardware unit produces an exception upon at least one of the stack-based instructions, and wherein the first unit will, in software, translate the at least one of the stack-based instructions causing the exception.

384. The system of claim 359, wherein the hardware unit is adapted to swap portions of an operand stack in and out of a register file of the first unit.

385. The system of claim 359, wherein the hardware unit includes an indication of the depth of a portion of operand stack stored in the register file.

386. The system of claim 359, wherein the hardware unit includes logic to keep a count of how many entries have been placed on an operand stack.

387. The system of claim 359, wherein the hardware unit includes logic that keeps track of a portion of a Java operand stack stored in a register file of the first unit.

388. The system of claim 387, wherein when a Java bytecode to be converted references an element of the Java operand stack stored in a register of the first unit's register file, and wherein the hardware unit produces an indication of that register to be used in the translation process.

389. The system of claim 359, wherein a register in a register file of the first unit contains the value for the top of the stack and wherein this register can change as a result of an executed instruction.

390. The system of claim 359, wherein the hardware unit stores the value of at least one Java register.

391. The system of claim 390, wherein the hardware unit stores the value of the OPTOP.

392. The system of claim 390, wherein the hardware unit stores the value of the Java Frame.

393. The system of claim 390, wherein the hardware unit stores the value of the Java VARS.

394. The system of claim 390, wherein the hardware unit stores the value of the Java PC.

395. The system of claim 390, wherein the hardware unit uses the at least one stored Java register value to construct at least some of the register-based instructions.

396. The system of claim 359, wherein a portion of the Java operand stack or Java variables are stored in the register file of the first unit, wherein the hardware unit keeps track of which registers in the first unit's register file contains the portion of the Java

operand stack or Java variables, the meaning of the registers being able to change as a result of an executed instruction.

397. The system of claim 359, wherein the stack-based instructions are Java bytecodes and wherein the hardware unit maintains at least one Java Variable in a register file of the first unit.

398. The system of claim 397, wherein the at least one Java Variable is stored in the register file separate from a portion of an operand stack.

399. The system of claim 397, wherein the hardware unit includes logic that keeps track of Java Variables stored in the first unit's register file.

400. The system of claim 399, wherein when a stack-based instruction to be converted references a Java Variable stored in a register of the first unit's register file, the hardware unit produces an indication of the register containing the Java Variable, the indication to be used in the conversion process.

401. The system of claim 399, wherein the first unit produces a "branch taken" indication and the hardware unit includes at least one pipeline stage that is flushed when the "branch taken" indication is produced.

402. The system of claim 401, wherein the hardware unit contains multiple pipeline stages, the pipeline stages being flushed when the "branch taken" indication is produced.

403. The system of claim 402, wherein the hardware unit includes a stack-based instruction buffer.

404. The system of claim 403, wherein the hardware unit includes a decode unit associated with the instruction buffer.

405. The system of claim 404, wherein decode unit is adapted to select multiple stack-based instructions to decode at one time.

406. The system of claim 359, wherein the first unit includes an execution unit to execute the register-based instructions.

407. The system of claim 359, wherein the first unit has two sets of register files.

408. The system of claim 407, wherein at least one set of register files are used by the first unit when converted instructions are provided by the hardware unit.

409. The system of claim 359, further comprising a multiplexer adapted to provide instructions from the hardware unit to the first unit or to provide instructions from another source to the first unit.

410. The system of claim 409, wherein the another source is a memory.

411. The system of claim 409, wherein the another source is the cache.

412. The system of claim 410, wherein the multiplexer is associated with a bus.

413. The system of claim 359, wherein a Java PC register and a native PC register are implemented using the same physical register.

414. The system of claim 359, wherein the hardware unit has access to at least one bus.

415. The system of claim 359, wherein the system has an indication used to switch to native mode.

416. The system of claim 415, wherein a portion of the operand stack is maintained in a register file of the first unit until a switch to the native mode.

D1 cont.

417. The system of claim 415, wherein the indication is stored in a register.

418. The system of claim 359, wherein under certain conditions, the hardware unit relinquishes control to a Virtual Machine running on the first unit.

419. The system of claim 359, wherein Java branch instructions are translated to register-based instructions in the hardware unit.

420. The system of claim 419, wherein a portion of the operand stack is maintained in a register file of the first unit while the Java branch instructions are translated to register-based instructions.

421. A method comprising:

storing stack-based instructions in a cache;

transferring at least one of the stack-based instructions to a hardware unit from the cache;

translating the at least one stack-based instruction from the cache into at least one register-based instruction in the hardware unit;

transferring the at least one register-based instruction from the hardware unit to a first unit; and

executing the at least one register-based instruction in the first unit.

422. The method of claim 421, further comprising transferring the stack-based instructions from a system memory to the cache.

423. The method of claim 421, wherein the cache is an instruction cache.

424. The method of claim 421, wherein the first unit and the hardware unit are part of a central processing unit.

425. The method of claim 421, wherein the first unit comprises a central processing unit and wherein the hardware unit is outside of the central processing unit.

426. The method of Claim 421, wherein the first unit, hardware unit and cache are on the same chip.

427. The method of Claim 421, wherein the stack-based instructions are Java bytecodes.

428. The method of claim 421, further comprising maintaining a portion of the operand stack is stored in a register file of the first unit.

429. The method of Claim 428, wherein the hardware unit produces register-based instructions that access the portion of the operand stack in the register file so as to reduce the number of register-based instructions that would otherwise be required.

430. The method of claim 428, wherein the at least one of overflow or underflow indications for the portion of the operand stack stored in the register file as produced by the hardware unit.

431. The method of claim 428, wherein an overflow or underflow produces operand transfer between the register file in the first unit and a memory.

432. The method of claim 430, wherein the registers of the register file of the central processing unit used to store the portion of operand stack is full of valid data.

433. The method of claim 430, wherein the at least one of the overflow or underflow indications is generated by a stack instruction pushing an operand or popping the operand from the operand stack.

434. The method of claim 428, further comprising maintaining the portion of the operand stack stored in the register file at or above a minimum level and/or at or below a maximum level between a maximum and a minimum level.

435. The method of claim 428, further comprising storing at least some Java registers in the register file.

436. The method of claim 421, wherein the hardware unit implements at least part of a Java virtual machine.

437. The method of claim 421, wherein the hardware unit is connected between a the cache memory and the first unit.

438. The method of claim 421, wherein the hardware unit is adapted to manage at least a portion of a Java stack.

439. The method of claim 421, further comprising in the hardware unit, examining the stack-based instructions to determine whether multiple stack-based instructions can be combined into fewer register-based instructions.

440. The method of claim 439, wherein the translating step is such that the hardware unit produces register-based instructions that access a portion of the operand stack in the register file so as to reduce the number of register-based instructions that would otherwise be required.

441. The method of claim 440, wherein multiple stack-based instructions pass through the hardware unit concurrently to allow for the operation of combining logic.

442. The method of claim 421, wherein the translating step is such that multiple Java bytecodes are converted into a single register-based instruction.

443. The method of claim 442, wherein the multiple Java bytecodes include a basic operand instruction and one or more stack manipulation instructions.

444. The method of claim 442, wherein the multiple Java bytecodes include a load or store instruction.

445. The method of claim 442, wherein the hardware unit produces an exception upon a stack-based instruction, and wherein the first unit will, in software, translate the exception-causing stack-based instruction.

446. The method of claim 421, further comprising swapping portions of an operand stack in and out of a register file of the first unit.

447. The method of claim 421, further comprising producing in the hardware unit, an indication of the depth of a portion of operand stack stored in the register file.

448. The method of claim 421, wherein the hardware unit includes logic to keep a count of how many entries have been placed on an operand stack.

449. The method of claim 421, wherein the hardware unit includes logic that keeps track of a portion of a Java operand stack stored in a register file of the first unit.

450. The method of claim 449, wherein when a Java bytecode to be converted references an element of the Java operand stack stored in a register of the first unit's register file, and wherein the translating step is such that the hardware unit produces an indication of that register to be used in the translation.

451. The method of claim 421, wherein a register in a register file of the first unit contains the value for the top of the stack and wherein this register can change as a result of an executed instruction.

452. The method of claim 421, wherein the value of at least one Java register is stored in the hardware unit.

453. The method of claim 421, wherein the hardware unit stores the value of the OPTOP.

454. The method of claim 421, wherein the hardware unit stores the value of the Java Frame.

455. The method of claim 421, wherein the hardware unit stores the value of the Java VARS.

456. The method of claim 421, wherein the hardware unit stores the value of the Java PC.

457. The method of claim 452, wherein the hardware unit uses the at least one stored Java register value to construct at least some of the register-based instructions.

458. The method of claim 421, wherein a portion of the Java operand stack or Java variables are stored in the register file of the first unit, wherein the hardware unit keeps track of which registers in the first unit's register file contains the portion of the Java operand stack or Java variables, the meaning of the registers being able to change as a result of an executed instruction.

459. The method of claim 421, wherein the stack-based instructions are Java bytecodes and wherein the hardware unit maintains at least one Java Variable in a register file of the first unit.

460. The method of claim 459, wherein the at least one Java Variable is stored in the register file separate from a portion of an operand stack.

461. The method of claim 459, wherein the hardware unit includes logic that keeps track of Java Variables stored in the first unit's register file.

462. The method of claim 456, wherein when a stack-based instruction to be translated references a Java Variable stored in a register of the first unit's register file, the hardware unit produces an indication of the register containing the Java Variable, the indication to be used in the conversion process.

D1 cont.

463. The method of claim 462, further comprising producing a "branch taken" indication by the first unit and finishing at least one pipeline stage of the hardware unit when the "branch taken" indication is produced.

464. The method of claim 463, wherein the hardware unit contains multiple pipeline stages, the pipeline stages being flushed when the "branch taken" indication is produced.

465. The method of claim 464, wherein the hardware unit includes a stack-based instruction buffer.

466. The method of claim 465, wherein the hardware unit includes a decode unit associated with the instruction buffer.

467. The method of claim 466, wherein decode unit is adapted to select multiple stack-based instructions to decode at one time.

468. The method of claim 421, wherein the first unit includes an execution unit to execute the register-based instructions.

469. The method of claim 421, wherein the first unit has two sets of register files.

470. The method of claim 469, wherein at least one set of register files are used by the first unit when translated instructions are provided by the hardware unit.

471. The method of claim 421, wherein the transferring of the at least one register-based instruction from the hardware unit to the first unit including passing through a multiplexer to provide instructions from another source to the first unit.

472. The method of claim 471, further comprising providing instructions from another source through the multiplexer to the first unit.

473. The method of claim 472, wherein the another source is a memory.

474. The method of claim 472, wherein the another source is the cache.

475. The method of claim 472, wherein the multiplexer is associated with a bus.

476. The method of claim 421, wherein a Java PC register and a native PC register are implemented using the same physical register.

477. The method of claim 421, wherein the hardware unit has access to at least one bus.

478. The method of claim 421, wherein the system has an indication used to switch to native mode.

479. The method of claim 421, wherein a portion of the operand stack is maintained in a register file of the first unit until a switch to the native mode.

480. The method of claim 421, wherein the indication is stored in a register.

481. The method of claim 421, wherein under certain conditions, the hardware unit relinquishes control to a Virtual Machine running on the first unit.

482. The method of claim 421, wherein Java branch instructions are translated to register-based instructions in the hardware unit.

483. The method of claim 482, wherein a portion of the operand stack is maintained in a register file of the first unit while the Java branch instructions are translated to register-based instructions.

484. A system comprising:

a first unit having a register file, the first unit adapted to execute register-based instructions; and

a hardware unit associated with the first unit, the hardware unit adapted to convert Java bytecode instructions into register-based instructions, wherein the hardware unit is adapted to store at least one Java register in the first unit's register file, the at least one Java Register including the Java Program Counter wherein the hardware unit receiving at least some stack-based instructions from a cache.

485. A system comprising:

a first unit adapted to execute register-based instructions, the first unit having an associated register file; and

a hardware unit associated with the first unit, the hardware unit adapted to convert stack-based instructions into register-based instructions, the hardware unit adapted to store portions of the operand stack in the first unit's register file, wherein the hardware unit is adapted to receive at least some stack-based instruction from a cache.

D1 cont.

486. A system comprising:

a first unit adapted to execute register-based instructions; and

a hardware unit associated with the first unit, the hardware unit adapted to convert stack-based instructions into register-based instructions, wherein certain instructions are not translated in the hardware unit but instead cause translation software to be executed by the first unit, wherein at least some of the stack-based instructions are from a cache.

487. A system comprising:

a first unit adapted to execute register-based instructions, the first unit having an associated register file;

a hardware unit associated with the first unit, the hardware unit adapted to convert stack-based instructions into register-based instructions; and

a multiplexer adapted to provide an instruction stream from a hardware unit or another stream from another source wherein at least some of the stack-based instructions come from a cache.

488. A system comprising:

a first unit adapted to execute register-based instructions, the first unit having an associated register file; and

a hardware unit associated with the first unit, the hardware unit adapted to convert stack-based instructions into register-based instructions, wherein the system has an indication used to switch to native mode, wherein at least some stack-based instructions are from a cache.

489. A system comprising:

a first unit adapted to execute register-based instructions, the first unit having an associated register file;

a hardware unit associated with the first unit, the hardware unit adapted to convert Java bytecodes into register-based instructions, wherein Java branch instructions are translated to register-based instructions while a portion of the Java operand stack is maintained in the register file.

490. A system comprising:

a first unit adapted to execute register-based instructions, the first unit producing a "branch taken" indication; and

a hardware unit associated with the first unit, the hardware unit adapted to convert stack-based instructions into register-based instructions to be provided to the first unit for execution, the hardware unit including at least one pipeline stage that is flushed due to the "branch taken" indication.

491. The system of claim 490, wherein the hardware unit contains multiple pipeline stages, the at least one of the pipeline stages being flushed when the "branch taken" indication is produced.

492. A system comprising:

a first unit adapted to execute register-based instructions, the first unit having an associated register file; and

a hardware unit associated with the first unit, the hardware unit adapted to convert Java bytecodes into register-based instructions, wherein the hardware unit includes logic that keeps track of a portion of the Java operand stack stored in the first unit's register file and when a Java bytecode to be translated references an element of the Java operand stack stored in a register of the first unit's register file, the hardware unit produces an indication of that register to be used in the translation process, wherein the portion of the operand stack stored in the register file is maintained at or above a minimum level and/or at or below a maximum level

493. A system comprising:

a first unit adapted to execute register-based instructions, the first unit having an associated register file; and

a hardware unit associated with the first unit, the hardware unit adapted to convert Java instructions into register-based instructions, the hardware unit adapted to maintain a portion of the operand stack in the first unit's register file, wherein the Java PC register and the native PC register are implemented using the same register.

494. A central processing unit comprising:

an input adapted to receive stack-based instructions;

a register file adapted to be manipulated using register-based instructions, the register file adapted to store a portion of the operand stack; and

a hardware subunit adapted to convert stack-based instructions into register-based instructions, wherein the portion of the operand stack is maintained in the register file.

D1 Cont.

495. A system comprising:

a first unit adapted to execute register-based instructions, the first unit having multiple sets of register files; and

a hardware unit associated with the first unit, the hardware unit adapted to convert stack-based instructions into register-based instructions, wherein at least one set of register files are used by the first unit when converted instructions are provided by the hardware unit.

496. A system comprising:

a first unit adapted to execute register-based instructions, the first unit having an associated register file; and

a hardware unit associated with the first unit, the hardware unit adapted to convert stack-based instructions into register-based instructions, the hardware unit adapted to store portions of the operand stack in the first unit's register file, wherein the portion of the operand stack is maintained at or above a minimum level and/or at or below a maximum level, wherein some stack-based instructions are executed in the first unit in software.

497. A system comprising:

a first unit adapted to execute register-based instructions;

a hardware unit associated with the first unit, the hardware unit adapted to convert stack-based instructions into register-based instructions; and

a multiplexer adapted to provide an instruction stream from the hardware unit or another stream from another source, wherein after power-up the multiplexer is adapted to bypass the hardware unit and to provide instructions from the another source.

498. A system comprising:

a first unit adapted to execute register-based instructions; and

a hardware unit associated with the first unit, the hardware unit adapted to convert stack-based instructions into register-based instructions, the hardware unit having an instruction buffer to store at least one stack-based instruction.

499. A system comprising:

a first unit adapted to execute register-based instructions, the first unit having an associated register file; and

a hardware unit associated with the first unit, the hardware unit adapted to convert stack-based instructions into register-based instructions, wherein the hardware unit is adapted to store a portion of the operand stack in the register file, wherein upon a switch to a native mode, the operand stack in an external memory is made consistent with the portion of the operand stack stored in the register file.

500. A system comprising:

a first unit having a register file, the first unit adapted to execute register-based instructions; and

a hardware unit associated with the first unit, the hardware unit adapted to convert Java bytecode instructions into register-based instructions, wherein a portion of the operand stack is stored on the first units register file and wherein the hardware unit is adapted to store at least one Java variable in the first unit's register file at a location separate from the portion of the operand stack, wherein at least one of the register-based instructions reference a register in the first unit's register file containing one of the at least one Java variable.

501. A system comprising:

a first unit adapted to execute register-based instructions, the first unit having an associated register file; and

a hardware unit associated with the first unit, the hardware unit adapted to convert Java bytecodes into register-based instructions, wherein a portion of the Java

operand stack is stored in the register file of the first unit, wherein the hardware unit keeps track of which registers in the first unit's register file contains portions of the Java operand stack, the meaning of the registers being able to change as a result of an executed instruction wherein the hardware unit maintains the portion of the operand stack at or above a minimum level and/or at or below a maximum level.

502. A system comprising:

a central processing unit having a register file, the central processing unit adapted to execute register-based instructions; and

a hardware unit associated with the central processing unit, the hardware unit adapted to convert stack-based instructions into register-based instructions, wherein a portion of the operand stack is stored in the register file of the central processing unit and wherein the hardware unit is adapted to produce at least one of overflow or underflow indications for the portion of the operand stack stored in the register file.

503. A central processing unit comprising:

an input adapted to receive Java bytecode instructions;

a register file adapted to be manipulated using register-based instructions; and

a hardware subunit adapted to convert Java instructions into register-based instructions, wherein the hardware subunit is adapted to store at least one Java register in the register file, the at least one Java Register including the Java Program Counter wherein the hardware unit is adapted to receive at least some Java instructions from a cache.

504. A system comprising:

an execution unit associated with a register file, the execution unit adapted to execute decoded instructions; and

hardware adapted to receive Java bytecodes and native non-Java instructions and adapted to produce decoded instructions to the execution unit, wherein a portion of the operand stack is stored in the register file, wherein at least some of the Java instructions are from a cache.

505. A system comprising:

a first unit adapted to execute register-based instructions, the first unit having an associated register file; and

a hardware unit associated with the first unit, the hardware unit adapted to convert Java bytecodes into register-based instructions, wherein a portion of the operand stack is stored in the register file of the first unit, wherein the hardware unit stores a top of stack pointer and uses it to construct the register-based instructions.